同期と順序のみによる 不明で不整合な時間の処理方法の提案と ウィキ町史への適用

山口琢 (フリー)　山形巧哉 (北海道森町)　小林龍生 (有限会社スコレックス)
大場みち子 (公立はこだて未来大学)

概要：われわれの日常生活や調べごとでは，日時が不明だったり時間的な前後関係が不整合なできごとに遭遇する．従来，過去のできごとを知識データベースとして活用するようなシステムでは，不明・不整合な時間データを受け付けてこなかった．本稿では，このようなできごと処理の支援システムについて，住民主体のデジタル町史「ウィキ町史プロジェクト」の実装を例にして論じる．

Processing inconsistent and unknown time by synchronization and order: application to Wiki Town History Project

Taku Yamaguchi (Independent Researcher)
Takuya Yamagata (Town Office of Hokkaido Mori-town)　Tatsuo Kobayashi (Scholex Co., Ltd.)
Michiko Oba (Faculty of Systems Information Science, Future University Hakodate)

Abstract: In our everyday life, we handle events whose date and time are unknown and inconsistent each other. Previous systems which use past events as knowledge did not accept those unknown and inconsistent date and time. In this paper, the requirements for a system which handles those kind of events are discussed, referencing the system developed in Wiki Town History Project.

1. はじめに

人が日常生活や調査の過程で行う時間処理は緩やかである．日時が不明なできごとでも時間的に処理するし，前後関係が不整合でも，なお時間的に扱う．地域の歴史を調べたり，自分の人生を振り返るときも同様である．そして，このような情報処理の支援も，IT の守備範囲である．

プログラミング言語には日時型オブジェクトがあり，さらに最近は時間データを可視化するプログラム・ライブラリが充実してきたので，時間データを処理するシステムを容易に実現できるようになってきた．しかし，これらのライブラリやそれを使ったシステムでは，われわれの日常生活や人文学・社会科学での時間処理の実態に比べて，時間データに求める制約が強い．

本稿では，人の日常生活や人文学・社会科学に寄与するために，時間処理のもっと小さな要件を見つけることを目標として，「同時・同期」と「順序」のみに着目し，かつ時間不明や時間的不整合も受け入れるシステムを提案する．ウィキ町史 (ちょうし) プロジェクトの試作システム「ウィキ町史ビューア/エディタ」を使って，この要件を具体的に論じる．

2. 背景と課題

背景としてウィキ町史プロジェクトを紹介し，課題を述べる．

2.1 ウィキ町史プロジェクト

北海道森町の ICT ボランティア団体ハウモリ [*1] が，地域の年表作成を通して住民に郷土愛を深めてもらう目的で始めたのが，ウィキ町史プロジェクトである [1][2]．住民の郷土への関心を深めるために，地域や地元の名所のデジタル年表・沿革 (ウィキ町史) を作成したり，あるいは郷土と関連づけて自分史を作成したりするのが有効ではないかと考えている．Wikipedia タウン開催と合わせてウィキ町史を作り，

[*1] http://howml.org/

ウィキテキスト形式で出力してウィキペディアに掲載したりしている．[*2] ウィキ町史ビューア/エディタは，既に公開されていて，[*3] 基盤システムとしてハッカソンに提供されてもいる．[*4]

沿革・年表を作るということは，対象について調べて，調べたことの根拠を文献に求めて，できごとを取捨選択して時間順序で並べるという頭を使った作業である．「縄文遺跡」といった他の地域と共通するテーマを盛り込むことで，他地域との関連を年表で表現し，地域間で互いに関心をもつことよる相乗効果を狙ったりもする．紙の町史・市史から，右から左に書き写すといった単純エントリー作業ではない．

プロジェクトが作成する年表「ウィキ町史」は，IPA 共通語彙基盤の語彙を使い，[*5] 関連する他のできごとや人名などへのリンクを含んだ Linked Open Data である．これによって，データ的にも他の地域などと連携しやすくなることを狙っている．

2.2 同時・同期

時制データベース (temporal database) や尺度の研究によれば，尺度としての時間は区間の系列である．[3]．できごとは，ある時間区間に属するものとして測定される．「同時」とは「同じ時間区間で起きる」ことである．カレンダー・時計の区間系列が，日常感覚の区間や必要とされる研究上の着目点と異なるとき，「同時」について不便が生じる．

2.2.1 深夜番組の放送時間

「12/19(土) 深夜 26 時，12/20(日) 午前 2 時 On Air」といった TV 放送番組の案内をよく見かける．これは冗長で，コンピュータ的には「12/20(日) 午前 2 時」だけで十分正確である．このような表現がなくならないことは，深夜がどちらの「日」に属するのか，人の感覚に必ずしもマッチしていないことを示している．

2.2.2 札幌本道

内浦湾をはさむ北海道の森町と室蘭市とはゆかりが深い．札幌本道開通に伴って，両町が本道における海路部分の両端となり，港が整備されたことが発展の契機となったからである．

しかし，例えば Wikipedia の記事から単純に年表を作って並べても，この点のアピールが弱い．森村駅逓所の開設と室蘭港の開港が，カレンダーでは「同時 (同年)」とはならないからである．「札幌本道が開通したときに」という「同期」を，カレンダー・時計で表現できない．

2.3 時間不明なできごとと順序

カレンダー・時計と対応づけなくても，できごと間の順序を検討する・決めることは，できごとに対する時間処理といえる．できごとの相対的な順序を決める研究 (relative dating) が，考古学では行われる [4]．

2.3.1 自分史

もっと身近な例として，自分の人生を振り返ってみよう．印象的なできごとの中には，日時が不明だったり，推定される日時と順序が不整合なことがあるだろう．

最終的に年が特定され，できごと間の順序が明らかになるとしても，それまでの思い出す過程で年表編集・表示システムがデータを受け付けてくれないのは不便である．

2.3.2 「藪の中」

芥川龍之介の小説「藪の中」では，盗人 (多襄丸)，死んだ男，男の妻である女の 3 者の言い分が食い違い，最後まで真実が明かされない．

フィクションであり，できごとの日時は原理的に決まらないが，登場人物も読者も時系列でできごとを語り，読み取る．読者が感じる食い違いには，語られたできごとの順序関係が含まれる．

3. 先行研究・従来の技術

先行する研究や実装を検討するにあたって，できごとが「過去か，将来か」という観点と，時間処理して答えを出す主体が「人間か，コンピュータか」という観点で整理する．

3.1 不明，未定な時間

過去の時間不明なできごと，将来の時間未定なできごと，それぞれについて先行研究・実装を検討する．

3.1.1 過去のできごと

過去のできごとを集めて時間的知識データベースとして利用する研究が AI の分野で行われた．時間処理して答えを出す主体はコンピュータである [5]．そのデータベース中のできごとは，時間的な問い合わせに答えを出すために使われるので，回答に寄与することを前提にしている．不整合なできごとがデータ

[*2] 例えば https://ja.wikipedia.org/wiki/森町_(北海道)
[*3] ビューア https://cc-study.appspot.com/my-town-timeline
エディタ https://cc-study.appspot.com/tle
[*4] LOD チャレンジ 2015 http://lodc.jp/2015/concrete5/resource#foundation_10
[*5] https://www.ipa.go.jp/osc/kyoutsugoikiban

ベースに追加されようとするとエラーを表示し，受け入れるときには問題データとして印を付ける．不整合の影響範囲を調べてエラーを排除する手法などが検討されたが，時間不明・不整合に対して消極的である [6].

ある部分集合に before/after の「鎖」関係 (全順序) が設定されているならば，それを利用して時間間隔の計算量を減らすといった手法も検討された．できごとの部分集合を順序集合とみなす手法については後で述べる．

3.1.2 将来のできごと

将来のできごとについては，ユーザにできごとの時間設定を要求しないシステムが，一般的にある．広い意味でのプロジェクト管理ツールは，各タスクの時間の入力を必ずしも求めない．ToDo リスト，タスク管理，工程管理といった分野である．

個人向け ToDo リスト・ツールは，ToDo の期限が必須ではなく，ToDo を実行する予定日時も必須ではない．時間的に処理して答えを出す – 期限を再設定するなど – 主体は人である．

PERT(Program evaluation and review technique) 図の期待時間はシステムが計算するもので，人が入力するものではない．答えを出す主体はコンピュータである．

この分野では，人の時間処理を支援するために，他にもさまざまな工夫がされている．

xfy Planner の「山積みスケジューラ」リフィルは，個人向けのタスク管理ツールであり，タスクの所要時間 (見積もり・実績) を日に割り振る UI になっている．[*6] 開始・終了時刻は設定しないので，その日のいつタスクを遂行する/したのか判らないが，所要時間の見積もり/実績は時分の粒度で設定・管理される．

3.1.3 同期と時間区間

勤務シフトをやりくりするアプリケーションでは，時計やカレンダーの時間区切りではなくシフトの時間区切りで勤務管理できる．日付をまたがる夜勤や「13:10」といった端数にとらわれることなく，シンプルに処理できる．答えを出す主体は人で，コンピュータは勤務シフトをシンプルに扱う手助けをしている．

時を RDF のリソースを介して間接的にカレンダー・時計と対応づける研究・実装がある [7][8]．後で述べるウィキ町史ビューア/エディタの「IRI 年」も同様なので，具体的にはそこで説明する．

git のようなファイルのバージョン管理システムには，バージョン系列のブランチとマージ機能がある．バージョンの順序は時間の順序でもある．異なるブランチのバージョンについては順序関係はない．マージが同期点となる．このブランチ/マージの様子を図で可視化するツールもある．

3.2 不整合

われわれは共観福音書や「藪の中」を題材に研究を進めてきた [9][10][11]．文章の引用関係に諸説あったり，できごとの順序関係に矛盾があることを，システムとして「正常」データとして受け入れる研究は少ない．

$a \leqq b \wedge b \leqq c \rightarrow a \leqq c$

という推移律が成り立たないことを受け入れると，できごとの集合は前順序集合ですらなくなってしまう．

3.3 まとめ

過去のできごとについて，不明・不整合を積極的に受け入れる研究やシステムは少ない．将来のできごと同様に，答えを出すのは人であるとして，それを支援するシステムが未開拓であると言える．

4. 時間処理の要件と研究のアプローチ

要件として，受け入れるべき前提を仮に設定し，表示 (可視化) 機能と編集機能を試作し，実践への適用して評価，モデルへフィードバックするイテレーションの研究アプローチをとる．既に公開中の「ウィキ町史エディタ/ビューア」があるので，前提を設定し，それに基づいて「ウィキ町史エディタ/ビューア」機能を確認・評価するところからスタートする．ここまでが本稿の範囲である．

4.1 要件

次の前提を受け入れて，人に馴染みのある「年表の Look and feel」でできごと処理を支援できる．これが要件である．本の巻末などで見られる普通の年表を想定していて，罫線の有無をおくと，表組みになるのが分かりやすいであろう．これ以上の具体的な可視化方法や編集操作を，現段階では問わない．

4.2 前提

順序がまったく当てにならないというのでは，システムとしてできることが少なすぎるだろう．次の [a]〜[b] のように前提を強めても，「藪の中」の例を

*6 http://yamahige5.typepad.jp/blog/cat6457567/

年	デモ #01	デモ #02
(設定なし)		#02-1: 年が指定されていないできごと。
幕末	#01-1: 幕末のできごと。	#02-2: 幕末のできごと。
(設定なし)	#01-2: 年が指定されていないできごと。	
2013	#01-3: 「2013年」のできごと。	
(設定なし)	#01-4: 年が指定されていないできごと。	
(設定なし)		#02-3: 年が指定されていないできごと。
2014		#02-4: 2014年のできごと。
(設定なし)		#02-5: 年が指定されていないできごと。

図 1　時間不明なできごとを含む二つの年表の並行表示
Fig. 1　Synchronization of two timelines both contain events whose time are unknown.

札幌本道	(明治6年) 札幌本道（日本初の本格的馬車道）の開通に伴い駅逓所ができる。札幌本道は、函館〜森間は道路であったが、森〜室蘭間が海路とされた。また、同年10月にベンジャミン・スミス・ライマンにより鷲ノ木の地質調査が実施された。 関連記事	(明治5年) 8月、札幌本道の開通に伴い、元室蘭（現在の崎守町）に室蘭海関所の業務が開始され、室蘭港が開港する。
1873		(明治6年) 3月、元室蘭に官立室蘭病院ができる。

図 2　西暦年をまたがるできごとの同期
Fig. 2　Synchronization of events which occured over multiple years.

受け入れられると考えられる．

[a] できごとは，いずれかの年表 (timeline) に属する．

[b] 年表内のできごとは，互いに順序づけられている．年表は全順序集合である．

[c] 異なる年表に属するできごとは，比較不能であったり，推移律が成立しなくてもよい．すなわち，できごとの全体は前順序集合ですらない．

[a]〜[c] の前提をおいてもなお，[c] があるため，従来よりは緩い制約となっている．

5.　ウィキ町史エディタの実装

前記の前提と要件の観点で，ウィキ町史エディターをチェックする．[*7]

5.1　ウィキ町史エディタの処理

まず，ウィキ町史ビューア/エディタの特徴的な「並行表示」処理を説明する．不明・不整合を含まない通常の年表イメージをなるべく崩さないよう処理している．

5.1.1　数値年，IRI 年または匿名年

現在のウィキ町史ビューア/エディタは，日や時分秒を無視して年のみを処理する．年には西暦年 (と和暦 (予定))，「IRI[*8] 年」，または年不明なできごとに割り当てられる匿名年がある．

IRI 年はユニークな (すなわち IRI で識別できる) ものごとなら何を指してもよく，それを時 (年) として扱うものである．例えば，「http://ja.dbpedia.org/resource/幕末」や「http://ja.dbpedia.org/resource/札幌本道」である．以降では略して，末尾の「幕末」を書くことがある．「幕末」などと西暦年との対応表「日本史時代区

[*7] https://cc-study.appspot.com/tle

年	できごと
幕末	#05-1: 幕末のできごと。
2015	#05-2: 数値で年が指定されているできごと。
(設定なし)	#05-3: 年が指定されていないできごと。
2013	#05-4: 数値で年「2013」が設定されているが、年の値が上のできごとの年よりも小さく、逆転しているできごと。
(設定なし)	#05-5: 年が指定されていないできごと。
2020	#05-6: 未来「2020年」のできごと。

図 3　数値年の順序が前後した年表の単体表示
Fig. 3　Single views of a timeline in which the order of numerical years are reversed.

年表 #03

年	できごと
幕末	#03-1: 幕末のできごと。
時間1	#03-2: 「時間1」のできごと。
(設定なし)	#03-3: 年が指定されていないできごと。
時間2	#03-4: 「時間2」のできごと。
(設定なし)	#03-5: 年が指定されていないできごと。
2020	#03-6: 未来「2020年」のできごと。

年表 #04

年	できごと
幕末	#04-1: 幕末のできごと。
2012	#04-2: 2012年のできごと。
時間2	#04-3: 「時間2」のできごと。
(設定なし)	#04-4: 年が指定されていないできごと。
時間1	#04-5: 「時間1」のできごと。
2020	#04-6: 2020年のできごと。

図 4　年表 #03 と#04 それぞれの単体表示
Fig. 4　Single views of timeline #03 and #04

年	年表 #03	年表 #04
幕末	#03-1: 幕末のできごと。	#04-1: 幕末のできごと。
時間1	#03-2: 「時間1」のできごと。	#04-5: 「時間1」のできごと。
(設定なし)	#03-3: 年が指定されていないできごと。	
時間2	#03-4: 「時間2」のできごと。	#04-3: 「時間2」のできごと。
(設定なし)	#03-5: 年が指定されていないできごと。	
2012		#04-2: 2012年のできごと。
(設定なし)		#04-4: 年が指定されていないできごと。
2020	#03-6: 未来「2020年」のできごと。	#04-6: 2020年のできごと。

図 5　年表 #03 と年表 #04 の並行表示
Fig. 5　Multi-view of timeline #03 and #04

[*8] Internationalized Resource Identifier，拡張国際化された URI (Uniform Resource Identifier)．URI も一般で使われる URL (Uniform Resource Locator) を拡張したものである．

分」がオープンデータ (CC-BY) として公開されていて，[*9] そこに該当する項目があれば，IRI 年は西暦年に置き換えて処理される．他のデータをこのような対応表として使うことは，技術的に容易である．

匿名年は，プログラムの処理の都合で，年不明な各できごとに割り当てられるユニークな (anonymous individual) 年である．画面上では年として何も表示されないが，後述の並行表示処理の説明で使う．

5.1.2 並行表示と同期

2 つ以上の年表を並べて表示する「並行表示」処理の概要を説明する．並行表示では，複数の年表が，1 つの「メイン」年表とそれ以外の「参考町史」年表とに区別される．メイン年表での順序を保つことが優先され，参考町史のできごとの順序は必ずしも保たれないことがポイントである．

並行表示は大まかに，次のアルゴリズム [#1]〜[#3] で生成される．

[#1] まず，メイン年表から「年」目盛りを生成する．「年」目盛りは，メイン年表でのできごとの順に，できごとの「年」を並べたものである．「年」の数値の順と異なることがある (できごとの順序優先)．

[#2] 次に「参考町史」年表中のできごとから「年」を取り出して，「年」目盛りに挿入する．この挿入処理では，メイン年表での年の順序を優先する．メイン年表のものと数値が同じ西暦年や同じ IRI 年は同一視する．この同一視によって複数の年表のできごとが「同期」して表示される．同一視されることで，順序が引き戻されたり飛ばされたりすることがある．また，前述の匿名年は，他の年と決して同一視されない．参考町史の IRI 年は，西暦年との対応表に該当しなければ，参考町史での順序を保つように挿入される．匿名年も同様に，参考町史での順序を保つように挿入される．以上の結果，参考町史から挿入された「年」の順序は，元の参考町史での順序を必ずしも保たない．

[#3] 最後に，年目盛りが完成してから，メイン年表と参考町史に左から列を割り当てて，できごとを対応する年の行に表示する．年目盛りにおいて参考町史の年は順序が保たれていないので，参考町史内のできごとの順序も保たれない．

5.2 同期と順序

図 1 では，2 つの年表が並行表示されている．それぞれの年表に，年不明のできごとがある．「#01-N」の N は各年表内での順序番号である．「幕末」で同期しているが，各年表内でできごとの順序が保たれている．年目盛りの「(設定なし)」は，プログラム内では匿名年に該当する．

図 2 の左側はメイン年表で森町の年表，右側は参考町史で室蘭市の年表である．駅逓所開設と室蘭港開港が IRI 年「札幌本道」で同期している．

5.3 不整合

年の数値が前後した年表 (カレンダーとの不整合) については，単体で表示するときは何も問題ない – ユーザが指定した順序で表示される．図 3 では，ユーザは#05-N の N の順でできごとを配置し，その通りに表示されている．ユーザ指定の順序を優先するため，年「2015」「2013」の順序が逆になっている．ここで，色を反転するなどして数値年の逆転を示すことも可能だ．しかし，勝手に並べ替えはしない．実際，並べ替えようとしても，#05-3 をどう処理すれば良いか判らない．

他の年表と並行表示した場合，起きている事態を直感的に示すのは難しくなる．図 4 は，2 つの年表#03 と#04 を，それぞれ単独で表示した場合を示している．「時間 1」という年と「時間 2」という年が同じ年を指すとしても，西暦何年なのか判らなければ，それぞれ別々に見る限りは問題ない．この 2 つの年表を，図 5 のように並行表示したとき，「時間 1」年と「時間 2」年とで同期しようとすると，参考町史の側でできごとの順序を保てない．#04-N の N の順序通りに並んでいないことが分かる．この#04-N がなければ，順序が変わったことに気づくのは難しいであろう．

6. 評価

6.1 機能的な評価

できごとの全体が全順序集合ならば，以上の実装によって，通常期待されるように年表を表示できる．

図 1 では，それぞれの年表内で順序が保たれているものの，年表をまたがった 2 つのできごとは，同期しているできごと以外は比較不能である．ところが，左側の#01-4 と右側の#02-3 は，あたかも#02-3 の方が後で起きたかのように表示されている．

また，図 5 の場合，このままでは，何が起きているかをユーザが理解するのは難しい．以上より，不整合がある場合については，時間的な処理を支援できるとは考えにくい．

[*9] http://linkdata.org/work/rdf1s2678i

図 6 年表 #03 と年表 #04 のクロス表示
Fig. 6 Cross view of timeline #03 and #04

7. 考察

7.1 年表間の不整合のクロス表示

図 5 のような不整合については，図 8 のようにクロスに可視化する実装が考えられる．縦軸が年表 #03 の時系列，横軸が年表 #04 の時系列であり，同期されたできごとが縦横軸の年に該当するセルに抜き出されている．順序に問題がなければ，できごとは年の進行に従って右肩下がりに配置される．不整合があれば，その部分で右肩上がりになる．年表#04 のできごとは，図中でオレンジの一重線で示される．左から右に追えば，上下はするものの，年表#04 の時間軸にそって追うことができ，かつ上に上がったところに不整合があると分かる．年表#03 のできごとは，図中で青の二重線で示され，たどり方は同様である．

図 7「藪の中」の男 (夫) と女 (妻) の証言から，順序を保ったまま抜粋したできごとを，クロス表示したものである．横軸が男 (夫，死霊) の証言の時間軸，

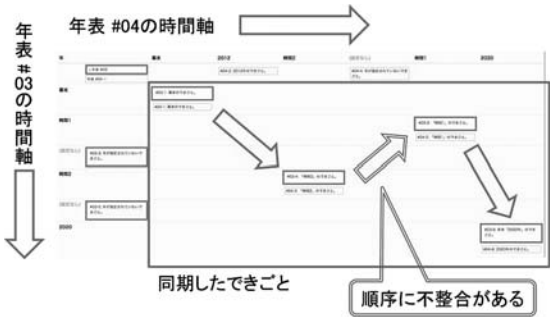

図 8 年表 #03 と年表 #04 のクロス表示が示す不整合
Fig. 8 Inconsistency indicated in the cross view

縦軸が女 (妻) の証言の時間軸である．「男が女を見た」，「男が身悶えした」，「多襄丸 (盗人) が女を蹴倒す」，「刀が男に刺さる」の起きた順番が，両者で異なることが分かる．

この表示方法では，時間軸を直線で示す点で，従来の年表のイメージを残しているが，画面の使用効率が悪く，大きなディスプレイでも全体を把握しきれない．

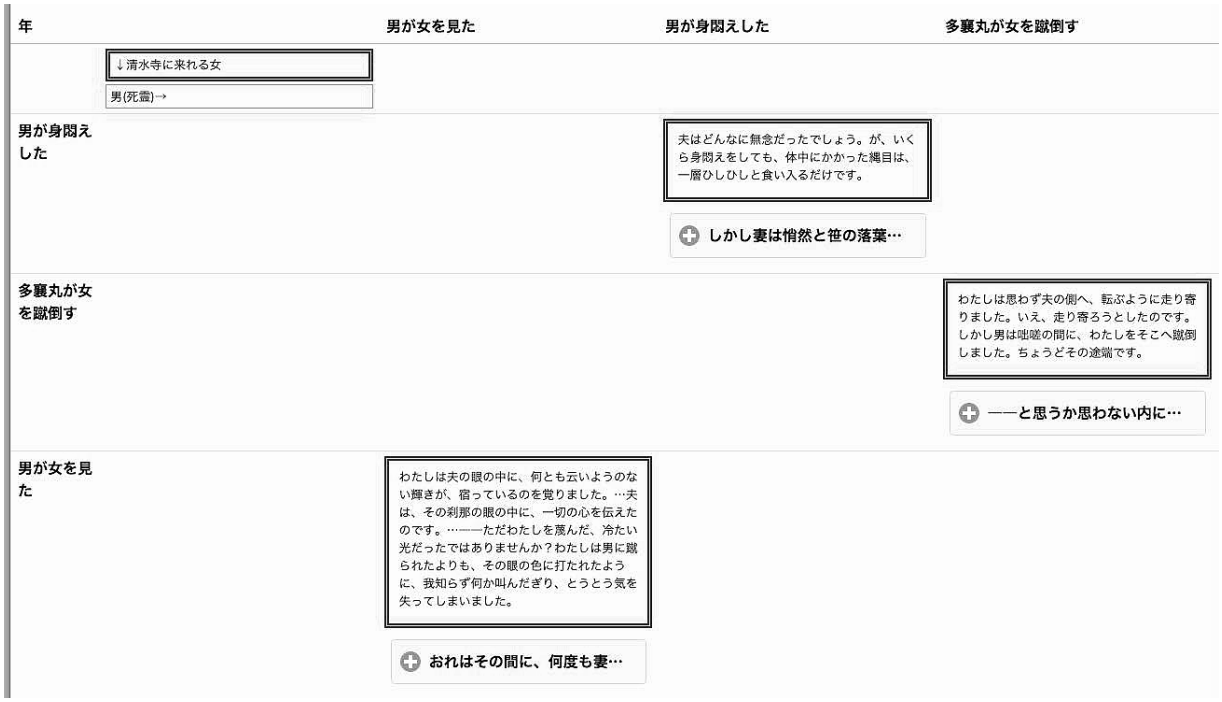

図 7 「藪の中」男と女の証言するできごとのクロス表示
Fig. 7 Cross view for "In a Grove"

また，この方法は 2 つの年表に対しては有効だが，3 つ以上を同時に表示できない．さらに工夫が必要となる．

7.2 半順序集合の年表

現ウィキ町史ビューア/エディタの手法では，年表内を全順序集合，すなわちどの 2 つのできごとをとっても順序が定まるという前提 [b] を採用した．これを受け入れると，年表を分けて作るときに，順序が分からないできごとを外出しにしなくてはならない．これでは，例えば，自分の歴史と地域の歴史といった自然な分け方で年表を作り分けることを妨げてしまう．

7.3 バージョン管理の方式

表組みをあきらめて，「3.1.3 同期と時間区間」で述べたファイルのバージョン管理システムが採用しているようなグラフを採用すると，評価で指摘した「年表をまたがった，できごとの順序の誤認」を緩和できるのではないか．ツリーではなく有効 (時間順序) 無閉路グラフとすることで，同期を扱えると考えられる．ただし，この場合でも閉路にあたる不整合は扱えない．

8. まとめ

従来の「まず時間があって，そこでできごとが生起する」という考え方に加えて，本研究が提案するのは「まずできごとがあって，できごとの順序と同時・同期関係を決めていくことで，時間が成立していく」という発想である．そのうえで，部分的な順序や同期関係に基づいて，できごと時間処理を支援することを提案した．

8.1 要件と前提の評価

人に馴染みのある「年表の Look and feel」でできごと処理を支援できる，との要件を設定し，次の 3 つの前提をおいた：[a] できごとはいずれかの年表に属する．[b] 各年表は全順序集合である．[c] 異なる年表に属するできごとは，比較不能であったり，推移律が成立しなくてもよい．

ウィキ町史エディタの実装を調べると，[b] を前提とすることで，[c] にもある程度対応できた．しかし，一方で [b] 制約は厳しすぎるかもしれず，他方で [c] については，実装の工夫や，不整合のもっと具体的な内容整理が必要と考えられる．

一緒に表示された街

表示する街(列)を選択...

一緒に表示	森町 (北海道)	室蘭市 (北海道)	八雲町 (北海道)	函館市 (北海道)	外ヶ浜町 (青森県)
森町 (北海道)	357	15	2	8	0
千歳市 (北海道)	3	3	1	2	0
室蘭市 (北海道)	15	32	1	11	0
伊達市 (北海道)	2	2	1	1	0
洞爺湖町 (北海道)	1	2	1	1	0
八雲町 (北海道)	2	1	2	2	0

図 **9** 年表が一緒に表示された回数の集計表
Fig. 9 Co-ccurrence of timelines

8.2 今後: 利用行動からの評価

時間処理で答えを出す主体が人であり，コンピューは支援に回るという要件により，ユーザが使っている様子を測定し評価する必要がある．直前で「不整合がある場合は…支援できるとは考えにくい」と指摘したが，そのような場合は少ないかもしれない．あるいは，意外と使いやすいかもしれない．

エディタについては，そのような測定・評価はまだである．

ビューアについても，本稿の観点からは測定されていないが，利用状況は測定している．図 9 は，2 つの年表が一緒に表示された回数を集計したものである．これではまだ遠いが，今後，不明・不整合を含む年表に対するユーザ行動を分析することで，解決策が見つかるかもしれない．

参考文献

[1] 山形巧哉，山口琢，大場みち子：歴史資産活用に係るオープンデータの重要性とその活用方法としてのウィキ町史プロジェクト，情報処理学会，研究報告ドキュメントコミュニケーション（DC），2015-DC-98，4，(2015)

[2] ハウモリ: ウィキ町史プロジェクトとは？, http://howml.org/wikichoshi-project (accessed on 2015-11-12)

[3] Christian S. Jensen, Curtis Dyreson: The Consensus Glossary of Temporal Database Concepts - February Version, http://people.cs.aau.dk/ csj/Glossary/index.html

[4] Paul Bahn: Archaeology: A Very Short Introduction (Very Short Introductions), OUP Oxford; Second Edition 版 (2012)

[5] James F. Allen: Maintaining knowledge about temporal intervals, Communications of the ACM, Volume 26 Issue 11, pp.832-843, 1983

[6] Kenneth Kahn, G.Anthony Gorry: Mechanizing temporal knowledge, Artificial Intelligence, Volume 9, Issue 1, August 1977, pp.87108

[7] 関野樹，原正一郎：地域研究における時空間情報の活用，情報処理学会，研究報告人文科学とコンピュータ（CH） 2012-CH-95(10), pp.1-6, (2012)

[8] 関野樹 : Linked Data における日の取り扱い－時間に基づくデータ連携，情報処理学会，じんもんこん 2014 論文集 2014(3), pp.125-130 (2014)

[9] 小林龍生，鎌田博樹，山口琢 : 共観年代記に向けて，情報処理学会，研究報告ドキュメントコミュニケーション（DC），2015-DC-98，6 (2015)

[10] 小林龍生，山口琢 : ParallelNarratology 試論一ハイパーテキストにおける相互参照の観点から－，情報処理学会，情報処理学会研究報告デジタルドキュメント（DD） (2007)

[11] 山口琢，小林龍生，大場みち子 : 定性的で主観的で個人的な記録を活用するシステムの試作 ～ 時間情報を例に，情報処理学会，情報処理学会研究報告デジタルドキュメント（DD） (2008)